

IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A system, comprising:
a computer-based execution system including a processor and a memory;
a runtime system stored in and run on the execution system that includes
an escape analysis module to (1) determine which objects of a program can be
stack allocated under a closed-world assumption and (2) analyze which stack allocation is
invalidated due to an occurrence of an open-world feature;
a stack allocation module to stack allocate the objects determined by the escape
analysis module; and
a stack allocation recovery module to recover invalidated stack allocations back to
their original allocations in heap based on the analysis of the escape analysis module by
suspending all threads so that safe patching and compensation publication can be made, patching
each allocation site of the invalidated stack allocations back to its original allocation in the heap,
and performing compensation publishing where necessary.
2. (Currently Amended) The system of claim 1, wherein the runtime system further
~~comprising~~ comprises a main engine to, when the open-world feature occurs, invoke (1) the
escape analysis module to check which stack allocation is invalidated and to identify allocation
sites of the invalidated stack allocations, and (2) the stack allocation recovery module to recover
the invalidated stack allocations.
3. (Canceled).
4. (Canceled).

5. (Currently Amended) The system of ~~claim 4~~claim 1, wherein the stack allocation recovery module performs the patching by patching stack allocation instructions back to heap allocation instructions.
6. (Currently Amended) The system of ~~claim 4~~claim 1, wherein the stack allocation recovery module performs the compensation publishing by:
- enumerating all stack object references to objects allocated on a stack into a record set;
 - going through all stack frames and identifying allocation sites of the invalidated stack allocations within each frame;
 - in each invalidated allocation site, checking if there is any stack object allocated at this site by searching the record set, wherein objects found in the record set are added, as initial publishing candidates, into a compensation publication;
 - publishing the compensation publication to the heap beginning with the initial publishing candidates.
7. (Currently Amended) A computer-implemented method of permitting stack allocation in a program with open-world features, comprising:
- determining which objects of the program can be stack-allocated under a closed-world assumption;
 - stack allocating these objects based on the determination;
 - analyzing which stack allocation is invalidated due to the occurrence of an open-world feature;
 - recovering those invalidated stack allocations back to their original allocations in heap based on the analysis by suspending all threads so that safe patching and compensation publishing can be made, patching each allocation site of the invalidated stack allocations back to its original allocation in the heap, and performing compensation publishing where necessary.
8. (Canceled).

9. (Currently Amended) The method of ~~claim 8~~claim 7, wherein the patching each allocation site of the invalidated stack allocations further comprises patching stack allocation instructions back to heap allocation instructions.
10. (Currently Amended) The method of ~~claim 8~~claim 7, wherein the compensation publishing further comprises:
- enumerating all stack object references to objects allocated on a stack into a record set;
 - going through all stack frames and identifying allocation sites of the invalidated stack allocations within each frame;
 - in each invalidated allocation site, checking if there is any stack object allocated at this site by searching the record set, wherein the objects found in the record set are added, as initial publishing candidates, into a compensation publication; and
 - publishing the compensation publication to the heap beginning with the initial publishing candidates.
11. (Canceled).
12. (Currently Amended) The method of claim 7, wherein the analyzing which stack allocation is invalidated is performed when the open-world feature occurs.
13. (Currently Amended) The method of claim 7, wherein the recovering those invalidated stack allocations is performed after analyzing which stack allocation is invalidated.
14. (Currently Amended) A machine-readable medium having stored thereon sequences of instructions, the sequences of instructions including instructions which, when executed by a processor, causes the processor to perform:
- determining which objects of a program can be stack-allocated under a closed-world assumption;
 - stack allocating the objects base on the determination;

analyzing which stack allocation is invalidated due to an occurrence of an open-world feature;

recovering invalidated stack allocations back to their original allocations in heap based on the analysis by suspending all threads so that safe patching and compensation publishing can be made, patching each allocation site of the invalidated stack allocations back to its original allocation in the heap, and performing compensation publishing where necessary.

15. (Canceled).

16. (Currently Amended) The machine-readable medium of ~~claim 15~~claim 14, wherein the patching each allocation site of the invalidated stack allocations further comprises patching stack allocation instructions back to heap allocation instructions.

17. (Currently Amended) The machine-readable medium of ~~claim 15~~claim 14, wherein the performing the compensation publishing further comprises:

enumerating all stack object references to objects allocated on a stack into a record set;
going through all stack frames and identifying allocation sites of the invalidated stack allocations within each frame;

in each invalidated allocation site, checking if there is any stack object allocated at this site by searching the record set, wherein the objects found in the record set are added, as initial publishing candidates, into compensation publication; and

publishing the compensation publication to the heap beginning with the initial publishing candidates.

18. (Currently Amended) The machine-readable medium of claim 14, wherein the analyzing which stack allocation is invalidated is performed when the open-world feature occurs.

19. (Currently Amended) The machine-readable medium of claim 14, wherein the recovering those invalidated stack allocations is performed after analyzing which stack allocation is invalidated.

20. (Original) The machine-readable medium of claim 14, wherein the instructions are part of a runtime environment.

21. (Original) The machine-readable medium of claim 14, wherein the machine-readable medium is a memory within a computer system.